**Lab 3**

**Create a new workspace and use the PLAYERS script provided in eLearning. For each problem, enter your query and attach a screen shot of the result set. Each question is worth 2 pts.**

1. **Create sequence for the PLAYERS table to be used to increment the primary key. The sequence should start with the next number after the last player it that was used and increment by 1. Show this new sequence in the corresponding system catalog table (data dictionary table). Take a screen shot of the select statement for the data dictionary table.**

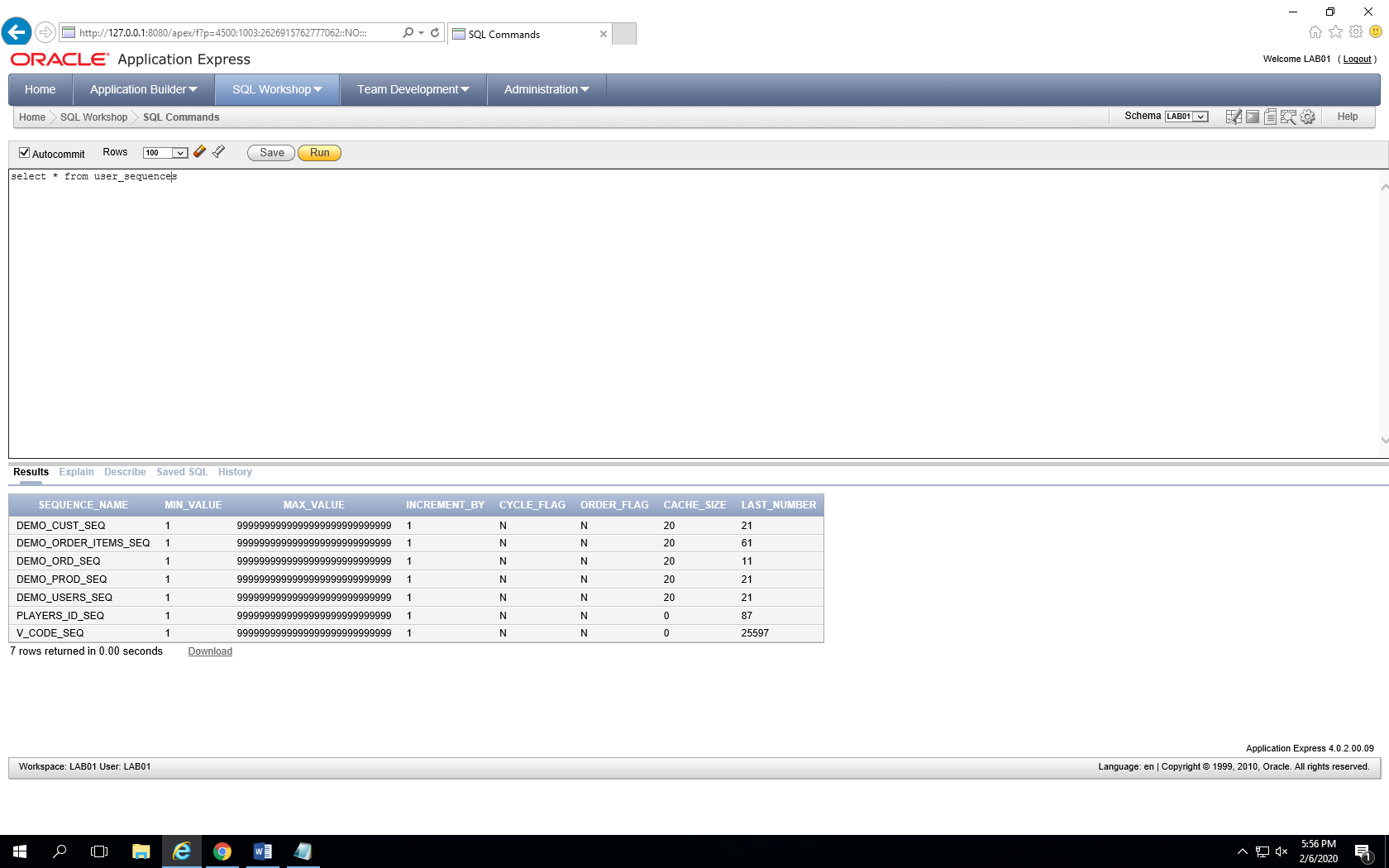
**SQL command to create the sequence:**

create sequence players\_id\_seq

start with 87 nocache

**SQL command to show that the sequence was created:**

select \* from user\_sequences



1. **Insert a new record into the PLAYERS table using the sequence. After you insert the row, write a select statement to select that new row. Provide a screen shot of the select statement.**

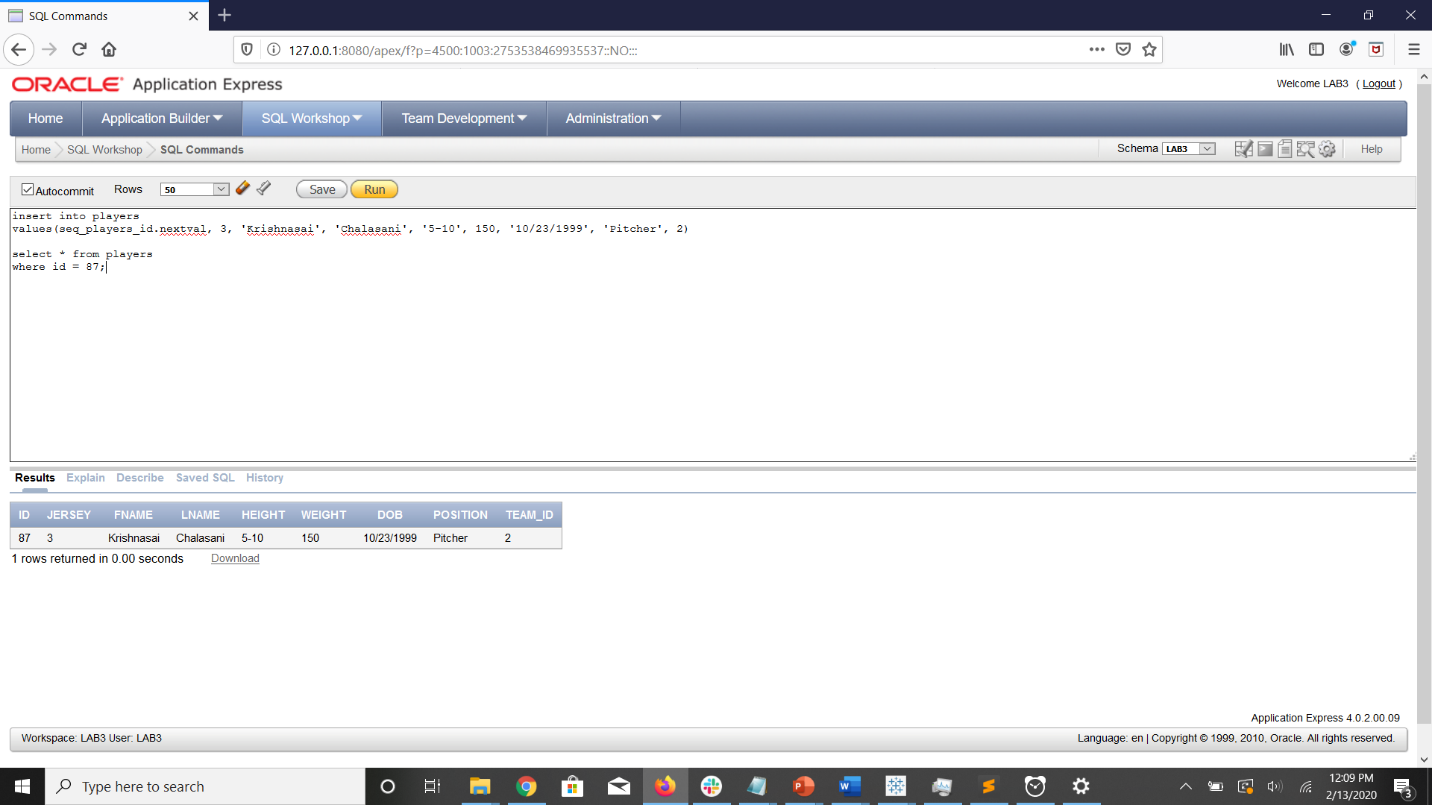
**SQL command to insert the new row:**

insert into players

values(players\_id\_seq.nextval, 3, 'Krishnasai', 'Chalasani', '5-10', 150, '10/23/1999', 'Outfielder', 2)

select \* from players

where id = 87

****

1. **Create a new table to track player workout data. The table should be setup with the following data types and constraints. Be sure to consider naming conventions.**

**Table Name:** ExerciseLog  
**Columns:**player id – pk, fk to player table  
date of exercise – pk, date  
weight - number(3), avoid nulls, make sure the value is > 100  
calories - number(3), avoid nulls  
exercise type – varchar2(50), avoid nulls  
time spent – number(3), avoid nulls

create table ExerciseLog(

player\_id constraint player\_id\_fk references players(id),

date\_of\_exercise date,

weight number(3) not null check(weight > 100),

calories number(3) not null,

exercise\_type varchar2(50) not null,

time\_spent number(3) not null,

primary key(player\_id, date\_of\_exercise)

)

1. **Add a column to the PLAYERS table that will be a number field (3). This column will indicate the leader for a particular position. You may name it lead\_id It will be the player id of the leader for that position. The lead\_id for all leaders will be null. After you add the column Update the data in the column as follows:**

Player id 21 is the lead for Catchers for Team 1(the lead\_id for player\_id 21 should be null).   
Player id 31 is the lead for Infielders for Team 1.  
Player id 32 is the lead for Outfielders for Team 1.  
Player id 9 is the lead for Pitchers for Team 1.  
Player id 63 is the lead for Catchers for Team 2.   
Player id 67 is the lead for Infielders for Team 2.   
Player id 77 is the lead for Outfielders for Team 2.   
Player id 43 is the lead for Pitchers for Team 2.

**It will look something like this:**



alter table players

add lead\_id number(3)

update players

set lead\_id = 21

where team\_id = 1

and position = 'Catcher'

and id != 21

update players

set lead\_id = 31

where team\_id = 1

and position = 'Infielder'

and id != 31

update players

set lead\_id = 32

where team\_id = 1

and position = 'Outfielder'

and id != 32

update players

set lead\_id = 9

where team\_id = 1

and position = 'Pitcher'

and id != 9

update players

set lead\_id = 63

where team\_id = 2

and position = 'Catcher'

and id != 63

update players

set lead\_id = 67

where team\_id = 2

and position = 'Infielder'

and id != 67

update players

set lead\_id = 77

where team\_id = 2

and position = 'Outfielder'

and id != 77

update players

set lead\_id = 43

where team\_id = 2

and position = 'Pitcher'

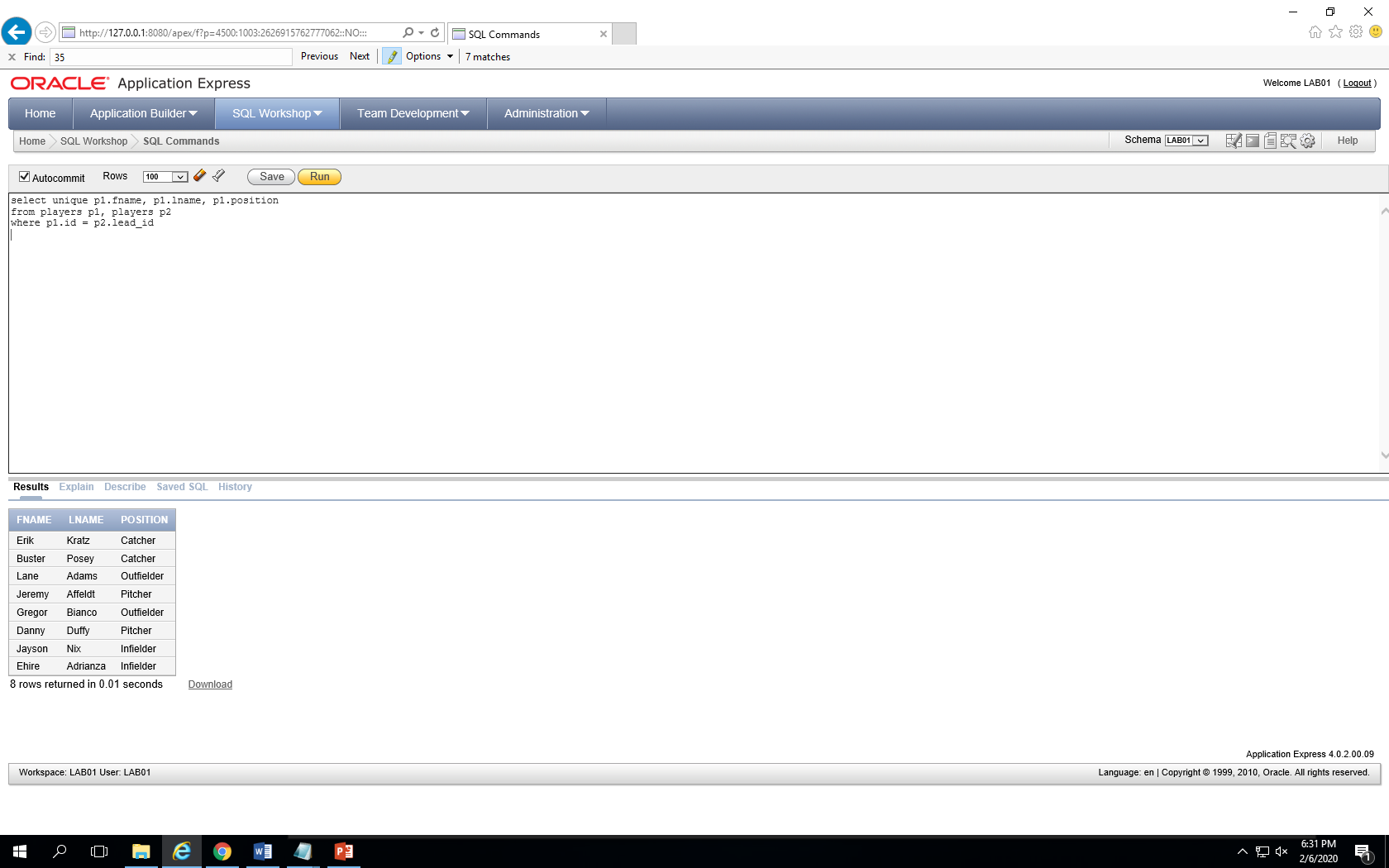
and id != 43

**Once completed, write a query that will list all the leader first names, last names, and position. Each leader name should be listed only once. (HINT: self join) Provide the query and screen shot.**

select unique p1.fname, p1.lname, p1.position

from players p1, players p2

where p1.id = p2.lead\_id

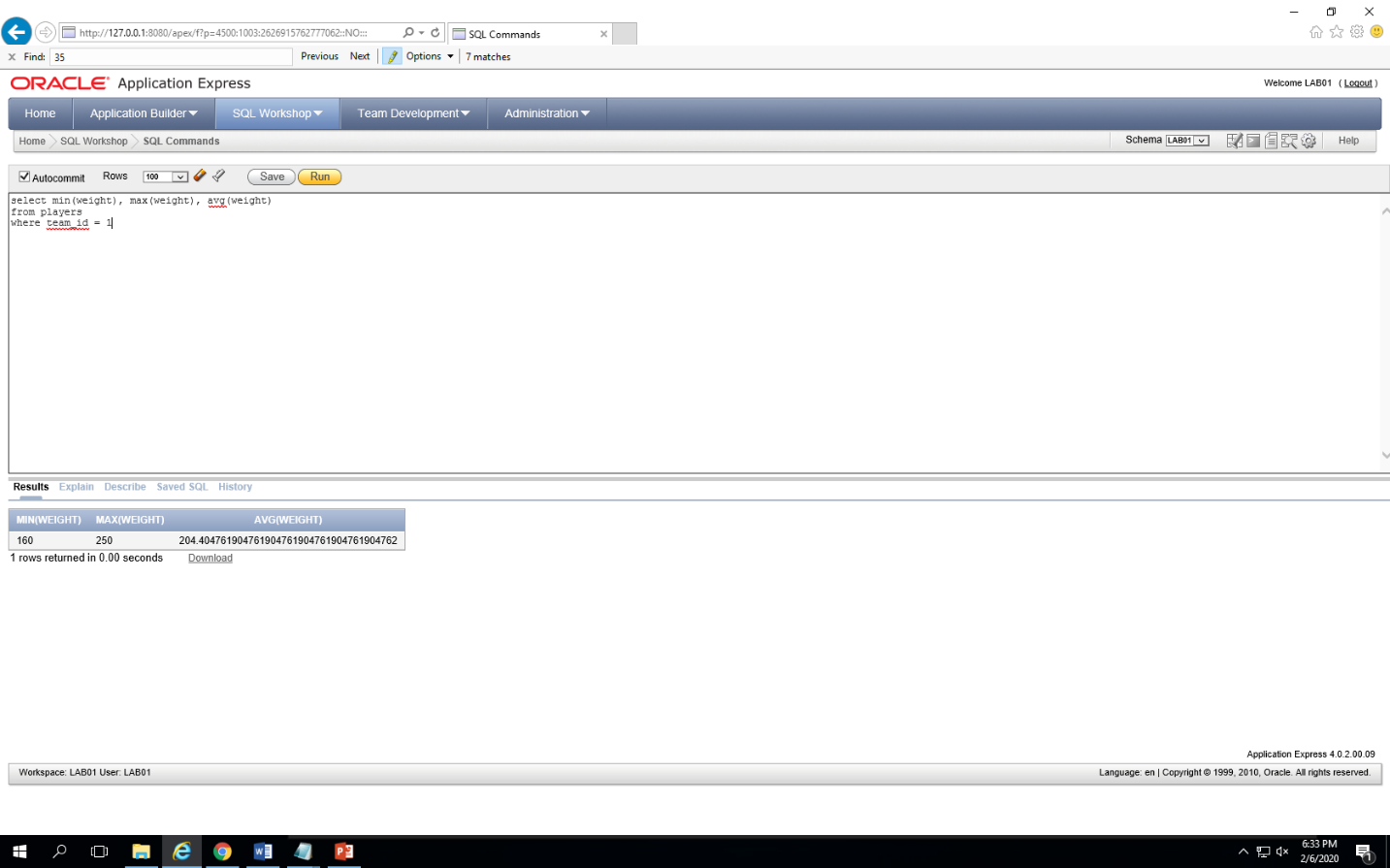


1. **What is the min, max, and average weight of the players on Team 1? Provide the query and screen shot**

select min(weight), max(weight), avg(weight)

from players

where team\_id = 1



1. **Remove Team 1 from the TEAMS table. Undo your changes. Now, remove all rows from the table, use the TRUNCATE command. Provide the queries below and the return message from each of the commands (e.g. 1 row deleted)**

delete

from players

where team\_id = 1

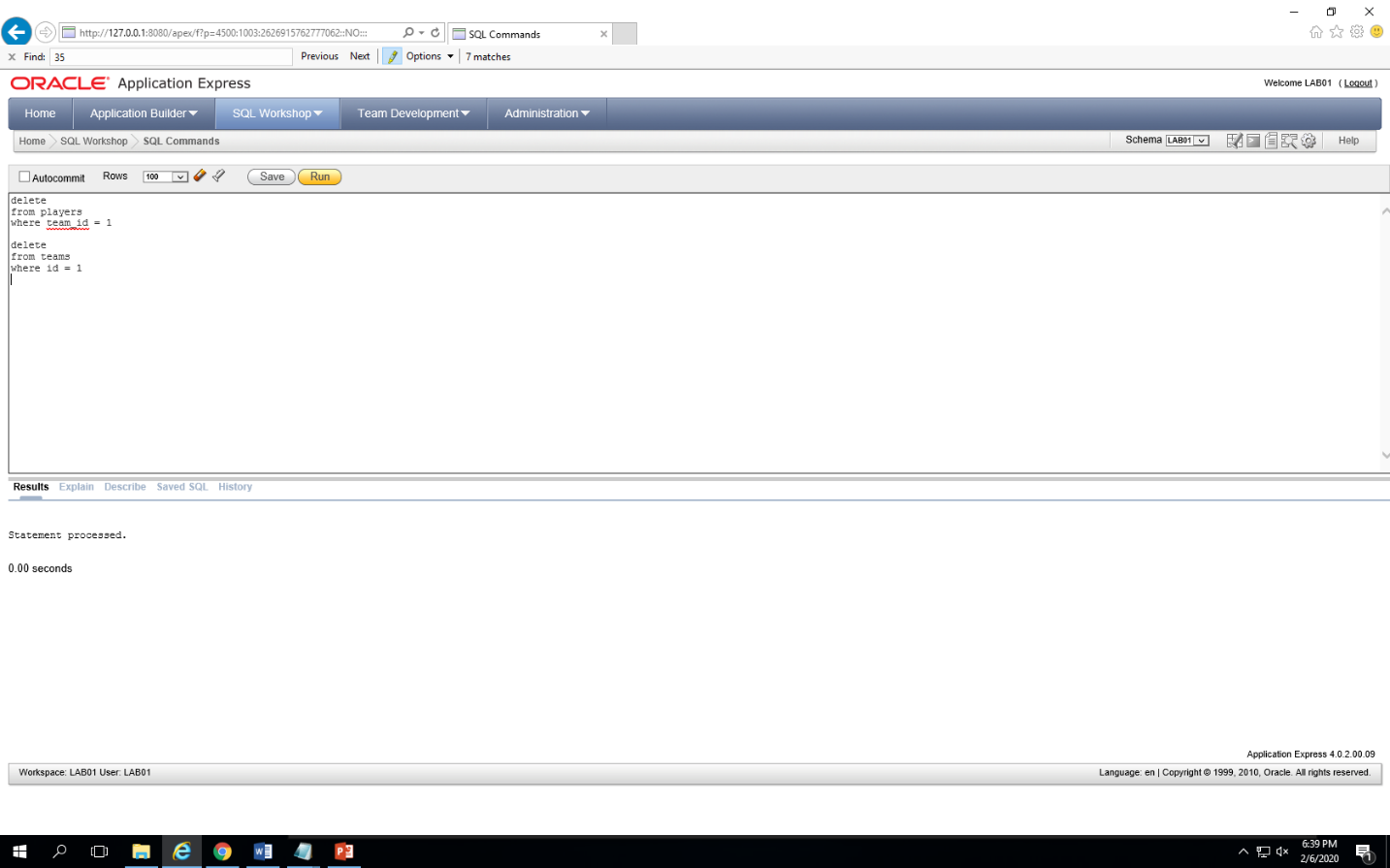
A screenshot of a computer

Description automatically generated

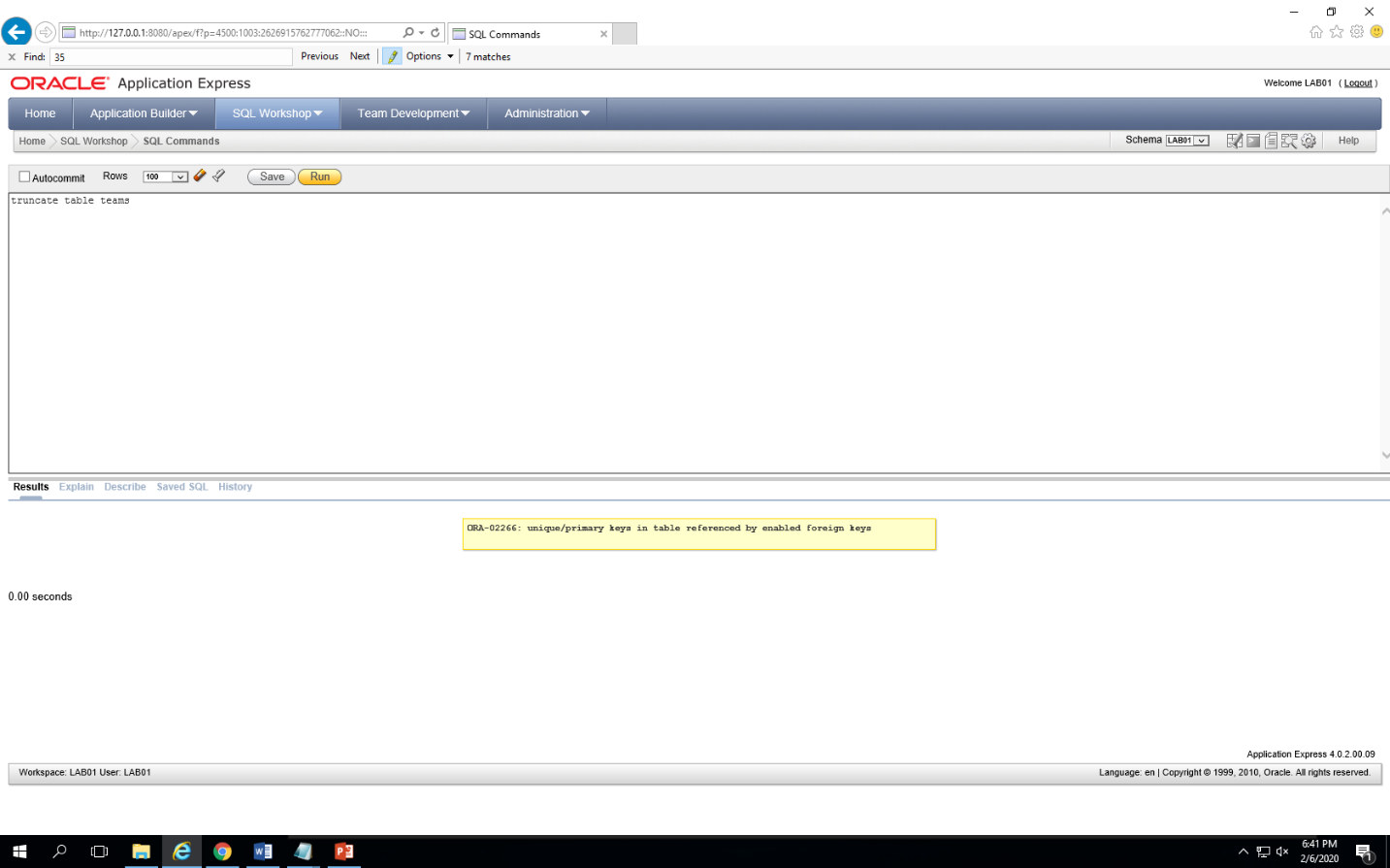
delete

from teams

where id = 1

****

truncate table teams

****